CLAIMS

1. (previously amended) A computer implemented system employing objects for generating an application script, in which both the objects and the script may be maintained separately, comprising:

a. means for wrapping objects with additional properties and events beyond those properties and events internal to the object; and

b. means for utilizing the additional properties and events to link and sequence the objects.

2. (previously amended) A computer implemented system employing objects for generating an application script, in which both the objects and the script may be maintained separately, representing a program structure comprising:

a. means for simultaneously displaying a plurality of different representations of the program structure; and

b. means for manipulating the program structure within each of the different representations

wherein the representations of the program structure may be synchronized.

3. (previously amended) The system of claim 2 further comprising a means for highlighting the icon for each object in the representations as objects are being instantiated during application development playback preview.

4. (previously amended) A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising:

a. a development environment and an interpreting run time environment; and

b. means for utilizing objects by specifying property values according to the script.

5. (previously amended) The system of claim 4 further comprising a means for communicating among objects through the exchange of property values.

6. (previously amended) The system of claim 5 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

7. (previously amended) The system of claim 4 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

8. (previously amended) A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising:

a. a development environment and an interpreting run time environment that have no logical or arithmetic operators; and

b. means for utilizing objects by specifying property values according to the script.

9. (previously amended) The system of claim 8 further comprising a means for communicating among objects through the exchange of property values.

10. (previously amended) The system of claim 9 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

11. (previously amended) The system of claim 8 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

12. (canceled) A computer implemented system employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising:

a. a development environment and an interpreting run time environment that have no definable data structure architecture; and

b. means for utilizing objects by specifying property values according to the script.

13. (previously amended) The system of claim 12 further comprising a means for communicating among objects through the exchange of property values.

14. (previously amended) The system of claim 13 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

15. (previously amended) The system of claim 12 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

16. (previously amended) The system of claim 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15 further comprising a means for adding additional programming constructs by employing objects that perform the function of programming constructs wherein unlimited expansion of program capabilities is achieved.

17. (previously amended) A computer implemented system employing objects and interpreting a script in which both the objects and the script may be maintained separately, comprising:

a. a run time program; and

b. means for utilizing objects according to the script.

18. (previously amended) A computer implemented system employing objects and interpreting a script in which both the objects and the script may be maintained separately, comprising:

a. a run time program that has neither arithmetic nor logical operators; and

b. means for utilizing objects according to the script.

19. (canceled) A computer implemented system employing objects and interpreting a script in which both the objects and the script may be maintained separately,, comprising:

a. a run time program that has no definable data structure architecture; and

b. means for utilizing objects according to the script.

20. (previously amended) A computer implemented development and run time system employing objects which utilizes a script, in which both the objects and the script may be maintained separately, utilizing a minimum set of core functionalities comprising:

a. means for instantiating objects;

b. means for integrating objects;

c. means for sequencing objects#; and

d. means for providing communication among objects

wherein the functionalities performed by the system during execution are determined by the objects used and the script.

21. (previously amended) A computer implemented run time system employing objects utilizing a minimum set of core functionalities which interprets a script, in which both the objects and the script may be maintained separately, comprising:

a. means for instantiating objects;

b. means for integrating objects;

c. means for sequencing objects; and

d. means for providing communication among objects;

wherein the functionalities performed by the system during execution are determined by

the objects used and the script.

22. (previously amended) A computer implemented system for employing objects, having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising:

a. means for setting the values of properties and connecting events;

b. means for recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and

c. means for traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

23. (previously amended) A computer implemented run time system employing objects which interprets a script containing property values and event settings, in which both the objects and the script may be maintained separately, and dynamically executes objects comprising:

a. means for wrapping objects with additional properties and events beyond those properties and events internal to the objects;

b. means for utilizing the additional properties and events to link and sequence the objects; and

c. means for reading one or more sets of property values and event settings maintained separately from the run time system and the objects

wherein the execution of the objects is determined by the property values and event settings in the script.

24. (previously amended) The system of claim 23 further comprising means for adding programming constructs or sub-languages utilizing objects.

25. (previously amended) A computer implemented system which interprets a script containing property values and event settings, which may be maintained separately, that distributes processing to objects, provides and manages data flow among objects, and manages the execution of objects comprising:

a. means for wrapping objects with additional properties and events beyond those properties and events internal to the object; and

b. means for utilizing the additional properties and events to link and sequence the objects

wherein the run time execution of the objects is determined by property values and events.

26. (previously amended) A computer implemented system employing objects which implements parallel processing comprising:

a. means for wrapping objects with additional properties and events beyond those properties and events provided internal to the object;

b. means for utilizing the additional properties and events to link and sequence the objects; and

c. means for specifying the temporal relationship among objects by placing the objects on one or more time lines

wherein execution of the objects occurs at least partially concurrently and during which property values may be exchanged among the objects and events may be initiated.

27. (previously amended) An object oriented programming computer implemented system in which the function of programming constructs is achieved by dynamically executing objects comprising:

a. means for wrapping objects with additional properties and events beyond those properties and events provided internal to the object;

b. means for utilizing the additional properties and events to link and sequence the objects; and

c. means for specifying a list of property values and event settings

wherein the execution of the objects is determined by the list of property values and event settings.

28. (previously amended) A computer implemented software method employing objects for generating an application script, in which both the objects and the script may be maintained separately, comprising the steps of:

a. wrapping objects with additional properties and events beyond those properties and events internal to the object; and

b. utilizing the additional properties and events to link and sequence the objects.

29. (previously amended) A computer implemented software method employing objects for generating an application script, in which both the objects and the script may be maintained separately, representing a program structure comprising the steps of:

a. simultaneously displaying a plurality of different representations of the program structure; and

b. means for manipulating the program structure within each of the different representations

wherein the representations of the program structure may be synchronized.

30. (previously amended) The software method of claim 29 further comprising the step of highlighting the icon for each object in the representations as objects are being instantiated during

application development run time preview.

31. (previously amended) A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising the steps of:

a. utilizing a development environment and an interpreting run time environment; and

b. utilizing objects by specifying property values according to the script.

32. (previously amended) The software method of claim 31 further comprising the step of communicating among objects through the exchange of property values.

33. (previously amended) The software method of claim 32 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

34. (previously amended) The software method of claim 31 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

35. (previously amended) A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising the steps of:

a. utilizing a development environment and an interpreting run time environment that have no logical or arithmetic operators; and

b. utilizing objects by specifying property values according to the script.

36. (previously amended) The software method of claim 35 further comprising the step of communicating among objects through the exchange of property values.

37. (previously amended) The software method of claim 36 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

38. (previously amended) The software method of claim 35 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

39. (canceled) A computer implemented software method employing objects and utilizing a script, in which both the objects and the script may be maintained separately, comprising the steps of:

a. utilizing a development environment and an interpreting run time environment that have no definable data structure architecture; and

b. utilizing objects by specifying property values according to the script.

40. (previously amended) The software method of claim 39 further comprising the step of communicating among objects through the exchange of property values.

41. (previously amended) The software method of claim 40 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

42. (previously amended) The software method of claim 39 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

43. (previously amended) The software method of claim 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, or 42 further comprising the step of adding additional programming constructs by employing objects that perform the function of programming constructs wherein unlimited

expansion of program capabilities is achieved.

44. (previously amended) A computer implemented software method employing objects and interpreting a script in which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

a. utilizing a run time program; and

b. utilizing objects according to the script.

45. (previously amended) A computer implemented software method employing objects and interpreting a script in which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

a. utilizing a run time program that has neither arithmetic nor logical operators; and

b. utilizing objects according to the script.

46. (canceled) A computer implemented software method employing objects and interpreting a script in which both the objects and the script may be maintained separately, for executing an application comprising the steps of:

a. utilizing a run time program that has no definable data structure architecture; and

b. utilizing objects according to the script.

47. (previously amended) A computer implemented development and run time software method employing objects for developing and executing an application which utilizes a script, in which both the objects and the script may be maintained separately, and utilizing a minimum set of core functionalities comprising the steps of:

a. instantiating objects;

b. integrating objects;

c. sequencing objects; and

d. providing communication among objects

wherein the functionalities performed by the software method during execution are determined by the objects used and the script.

48. (previously amended) A computer implemented run time software method employing objects for executing an application utilizing a minimum set of core functionalities which interprets a script, in which both the objects and the script may be maintained separately, comprising the steps of:

a. instantiating objects;

b. integrating objects;

c. sequencing objects; and

d. providing communication among objects;

wherein the functionalities performed by the software method during execution are determined by the objects used and the script.

49. (previously amended) A computer implemented software method for employing objects, having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising the steps of:

a. setting the values of properties and connecting events;

b. recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and

c. traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

50. (previously amended) A computer implemented run time software method employing

objects which interprets a script containing property values and event settings, in which both the objects and the script may be maintained separately, and dynamically executes the objects comprising the steps of:

a. wrapping objects with additional properties and events beyond those properties and events internal to the objects;

b. utilizing the additional properties and events to link and sequence the objects; and

c. reading one or more sets of property values and event settings maintained separately from the run time system and the objects

wherein the execution of the objects is determined by the property values and event settings in the script.

51. (previously amended) The software method of claim 50 further comprising the step of adding programming constructs or sub-languages utilizing objects.

52. (previously amended) A computer implemented software method which interprets a script containing property values and event settings, which may be maintained separately, that distributes processing to objects, provides and manages data flow among objects, and manages the execution of objects comprising the steps of:

a. wrapping objects with additional properties and events beyond those properties and events internal to the object; and

b. utilizing the additional properties and events to link and sequence the objects

wherein the execution of the objects is determined by the property values and events.

53. (previously amended) A computer implemented software method employing objects which implements parallel processing comprising the steps of:

a. wrapping objects with additional properties and events beyond those properties

and events provided internal to the object;

b. utilizing the additional properties and events to link and sequence the objects; and

c. specifying the temporal relationship among objects by placing the objects on one or more time lines

wherein execution of the objects occurs at least partially concurrently and during which property values may be exchanged among the objects and events may be initiated.

54. (previously amended) A computer implemented object oriented software programming method in which the function of programming constructs is achieved by dynamically executing objects comprising the steps of:

a. wrapping objects with additional properties and events beyond those properties and events provided internal to the object;

b. utilizing the additional properties and events to link and sequence the objects; and

c. specifying a list of property values and event settings

wherein the execution of the objects is determined by the list of property values and event settings.

55. (previously added) A computer implemented software method providing a general solution for employing standardized objects with properties not internal to the standardized objects comprising the steps of:

a. wrapping standardized objects with additional properties beyond those properties internal to the standardized object; and

b. utilizing the additional properties to control the standardized objects.

56. (previously added) A computer implemented software method providing a general solution for employing standardized objects with events not internal to the standardized objects

comprising the steps of:

a. wrapping standardized objects with additional events beyond those events internal to the standardized object; and

b. utilizing the additional events to control the standardized objects.

57. (previously added) A computer implemented software method providing a general solution for employing standardized objects with properties and events not internal to the standardized objects comprising the steps of:

a. wrapping standardized objects with additional properties and events beyond those properties and events internal to the standardized object; and

b. utilizing the additional properties and events to control the standardized objects.